ABSTRACT

A programmable interpreter for creating, interpreting, and executing a programming language. The present invention is a virtual processor that eliminates interpretation of pseudo code typical of common interpretive engines. The preferred embodiment of the present invention includes a computer system comprising a bus communicating information, a processor, and a random access memory for storing information and instructions for the processor. The processing logic of the preferred embodiment is operably disposed within the random access memory and executed by the processor of the computer system. A command stream, comprising a command identifier or function name in combination with a string of arguments, is a typical input for the processing logic of the present invention. Upon activation of the processing logic of the present invention, a parser is executed to manipulate the input command stream and produce an execution stream with a processing component identifier corresponding to the specified command. The command is then executed indirectly and a pointer is updated to point to the next command in the execution stream. Arguments for commands are pushed on to and popped from a stack. Results from the execution of commands are pushed onto a stack. For commands that define a new function or procedure, frame data is maintained to preserve the context in which the new function or procedure is executed. Each command in the stream is interpreted in this manner until the end of the execution stream is reached.